SAE 3.05

CRYPTO

Romain Mechain Groupe 21B

Arthur VILLETTE Groupe 23B

Echauffement:

indice 1:

Pour déchiffrer l'indice 1 nous avons regardé le message crypté et nous avons vu qu'une lettre apparaissaient plus souvent que les autres, la lettre 'q', donc on fait un fonction pour s'en assurer. Ensuite on sait que la lettre la plus courante est 'e', donc on a fait une fonction

de décalage entre 'q' et 'e' pour savoir de combien on va décaler les autres lettres selon la méthode de césar, car on pense que le message a été chiffré avec le chiffrement de césar. En utilisant notre fonction on trouve un message clair.

Le message trouvé est donc : *PRES DU CHEMIN SE CACHE UN TRESOR ACCROCHE A UN ARBRE TOUT RECOUVERT D'OR NE NEGLIGE PAS LA JEUNE POUCE FEUILLU GRAND EST SON SECRET MALGRE SA TAILLE MENUE RONDES ET COLOREES SONT LES BAIES QU'IL PORTE ANISEES ET SUCREES, LEURS SAVEURS SONT FORTES. MAIS ATTENTION A NE PAS LES CROQUER, MEME SI LA FAIM TIRAILLE TES ENTRAILLES, EN AUCUN CAS TU NE DOIS SUCCOMBER*

clef 1:

Cependant, la clé ne pouvait pas être tout le message, nous avons donc pensé qu'un autre message était caché dans celui-ci. Nous avons donc essayé de prendre la première lettre de chaque ligne, et en effet, c'était ce qu'il fallait faire.

Le message était donc : PANGRAMME

Indice 2:

Au vue du message numéro 2, et de la réponse du précédent, nous avons pensé que la clé devait être le message caché dans le message 1. Et grâce à cela nous avons déterminé qu'il s'agissait d'un cryptage de vigenère.

En décryptant le message 2 nous avons obtenue : `LE VYF ZEPHIR JUBILE SUR LES KUMQUATS DU CLOWN GRACIEUX\nIL CACHE DANS LA REPETITION LE SECRET DE CES MURMURES MALHEUREUX\nNE GARDEZ DU PREMIER SOUFFLE QUE LES PREMIERES APPARITIONS\nET AINSI DEVOILEZ LE MESSAGE CACHE DERRIERE LA SUBSTITUTION

Clef 2:

Ici, la clé était cachée par une énigme, le premier souffle correspond à la première ligne du message, et on indique de garder uniquement la première apparition de chaque lettre. Comme on nous parle de substitution, il suffisait de créer une clé de substitution, avec les premières occurrences.

On obtient : ABCDEFGHIJKLMNOPQRSTUVWXYZ → LEVIFZPHYRJUBSKMQATDCOWNGX

réponse:

Enfin, il ne reste plus qu'à appliquer la substitution.

On obtient : BRAVO, VOUS AVEZ GAGNEZ! LE CODE A FOURNIR EST: ELIZEBETH

Partie 1:

1) En supposant que RSA soit utilisé correctement, Eve peut-elle espérer en venir à bout? En vous appuyant sur votre cours, justifiez votre réponse.

L'algorithme RSA utilise des nombres premiers pour la création de ses clés, de ce fait, la taille de ces derniers est très importante. En effet, il est très compliqué de factoriser des grands nombres premiers en produit de deux autres nombres premiers. Ici, on nous indique que RSA est correctement utilisé, on suppose donc que des nombres de plus de 2048 bit sont utilisés, si c'est le cas, le temps de cassage par force brut serait suffisamment important pour considérer le chiffrement comme sûr.

2) En quoi l'algorithme SDES est-il peu sécurisé? Vous justifierez votre réponse en analysant le nombre d'essai nécessaire à une méthode "force brute" pour retrouver la clé.

Le SDES est peu sécurisé et même actuellement obsolète à cause de ses clés courtes qui font seulement 10 bit, ce qui rend le cryptage très fragile et facilement décryptable par un algorithme en force brut. Il n'y a en réalité que 1024 possibilité pour ce cryptage.
exemple de décryptage:

3) Est-ce que double SDES est-il vraiment plus sur? Quelle(s) information(s) supplémentaire(s) Eve doit-elle

   récupérer afin de pouvoir espérer venir à bout du double DES plus rapidement qu'avec un algorithme brutal?
   Décrivez cette méthode astucieuse et précisez le nombre d'essai pour trouver la clé

Le double SDES est bien réellement plus sur que le simple car le nombre de possibilité passe de 1024 à 1024*1024, cependant il n'en reste pas moins peu efficace car il ne résout pas le problème de la taille des clés qui sont toujours faibles, et possiblement décryptage en un temps raisonnable. Cependant il est possible considérablement diminuer le nombre

d’essaie en utilisant la technique du “meet in the middle”, consistant à crypter le message clair, et décrypter le message crypté, et voir si les résultats sont similaires. Le nombre d’essaie passe alors à 1024*2.

Partie 2:

1) Est-ce vraiment un problème? Justifiez votre réponse.

En tant que personne souhaitant décrypter le message, il s’agit en effet d’un problème. Jusqu’à présent, les clés étant limitées à 10 bit et donc 1024 possibilité, la trouver en utilisant un cassage brut était raisonnable, de plus, en utilisant un double cryptage, cela nous laisse la possibilité d’utiliser des méthodes permettant de limiter le nombre d'itérations à effectuer. Cependant, ce n’est pas du tout le cas avec l’AES, en effet, il s’agit d’un cryptage utilisant des clé de 256 bit, la méthode de cassage brutale n’est donc plus du tout envisageable, en effet il y a 2**256 clés possibles, le temps de tous les parcourir n’est plus raisonnable.

2) Temps de cassage AES

Pour casser AES il faudrait tester toutes les possibilités donc 2^256 et le multiplier par le temps de déchiffrage d’un essaie.

Exemple pour un message de taille 1500 qu’on crypte avec AES:

Le temps de décryptage du message est de:
d= 0.00011574399832170457
donc le temps cassage brutal:
cb= 2^256*d
cb= 1.34022398*10^73 seconde
cb=3,17097919837645876*10^65 anne
donc impossible à casser en brutal

3) Il existe d’autres types d’attaques que de tester les différentes possibilités de clés. Lesquelles? Vous donnerez une explication succincte de l’une d’elles.

Il existe en effet d'autres méthodes pour trouver une clé de chiffrement, sans avoir à tester toutes les possibilités. En effet, il y a des méthodes bien plus efficaces que le cassage par

force brute. Parmi les plus connus on retrouve notamment l'attaque d'analyse par fréquence, ou bien l'attaque de "Man-in-the-middle". Parlons plus précisément de cette dernière. Plutôt que de casser le chiffrement lui-même, un attaquant peut essayer de s'insérer entre les deux parties communicantes pour intercepter et potentiellement altérer les communications. Cependant, cela implique de s'appuyer sur l'ensemble des communications, et non seulement sur un message.

Explication images :

Pour trouver le code caché dans les deux images qui nous sont fournies, nous avons commencé par rechercher dans dans le code de la première. De ce fait, nous avons remarqué que l'ensemble de ces pixels était paires, ce qui nous a paru étrange. En regardant dans la deuxième, nous avons aussi remarqué que ce n'était pas le cas pour les premiers pixels de la deuxième. Nous nous sommes donc rappelé de notre SAE image de première année, ou nous avions fait quelque chose de similaire. Nous avons transformé la deuxième image en binaire en mettant les bits à 0 lorsqu'ils sont paires, et 1 sinon. De ce fait en récupérant les 64 premiers on obtient :
`1110011101101101001100010011111110010010101110011001000001001100`

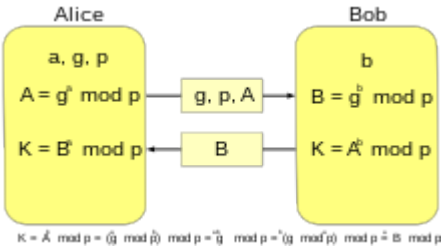
Partie 4:

1) Alice et Bob utilisent toujours la même clé. Est-ce une bonne pratique?

Utiliser toujours la même clé est une mauvaise pratique car si la clé d'un des messages est trouvée, alors tous les messages sont déchiffrables. Une bonne pratique serait donc d'utiliser une clef différente pour chaque message, ainsi, si un message est déchiffré, ce qui signifierait que la clé à été trouvée, les autres ne seront alors pas déchiffrables. Cependant cette solution n'est pas sans défaut, le problème d'utiliser une clef différente pour chaque message est de pouvoir communiquer les différentes clés entre chaque individu qui communique, ce qui implique un risque important.

2) Le protocole PlutotBonneConfidentialité est inspiré d'un vrai protocole réseau. Lequel? Décrivez la partie associée à la certification des clés qui est absente de PlutotBonneConfidentialité.

   Le protocole PlutotBonneConfidentialité décrit ici semble être inspiré du protocole Diffie-Hellman.
   certification des clés: fonctionnement:



1- Alice et Bob ont choisi un nombre premier **P** et **G** un générateur (La racines primitives modulo P)

2- Alice choisit un nombre aléatoire **a** et envoie a Bob **A=** g puissance a modulo p

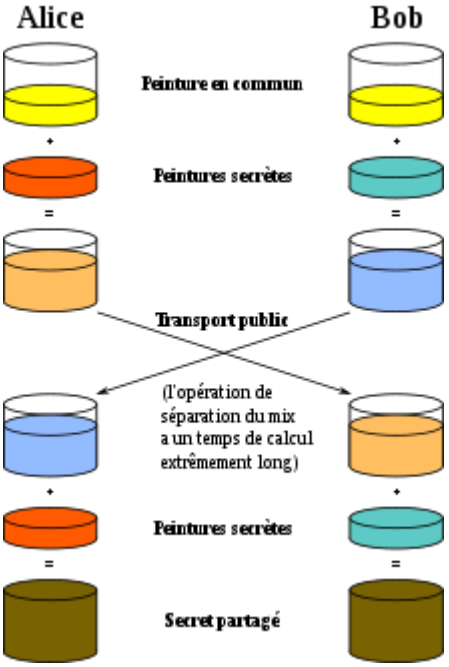
3- Bob choisit un nombre aléatoire **b** et envoie à Alice **B=** g puissance b modulo p

4- Alice utilise B et fais B^a[P] ce qui vaut **g^(ba)[P]**

5- Bob utilise A et fais A^b[P] ce qui vaut **g^(ab)[p]**

ducoup Alice et Bob on la même clef secrète

illustration de la certification des clés plus simple:



3) Il n'y a pas que pour l'échange de mots doux qu'un tel protocole peut se révéler utile. . . Donnez au moins deux autres exemples de contexte où cela peut se révéler utile.

Le cryptage peut se révéler utile dans plein de domaines, surtout ceux utilisant régulièrement des données sensibles.

- Le cryptage est largement utilisé dans le contexte militaire pour sécuriser les communications et protéger les informations sensibles contre toute interception par des adversaires ou des parties non autorisées. Les forces armées du monde entier utilisent des techniques de chiffrement avancées pour garantir la confidentialité, l'intégrité et l'authenticité de leurs communications.

- Dans le contexte bancaire, le cryptage est largement déployé pour protéger les données financières des clients, telles que les informations de carte de crédit, les identifiants personnels et les transactions. Lorsqu'un client accède à son compte en ligne, transmet des informations sensibles ou effectue une transaction, les données sont généralement cryptées avant d'être envoyées sur le réseau.

4) Connaissez-vous des applications de messagerie utilisant des mécanismes de chiffrement similaires? (on parle parfois de chiffrement de bout en bout)? Citez-en au moins deux et décrivez brièvement les mécanismes cryptographiques sous-jacents.

on a comme messagerie utilisant des mécanismes de chiffrement de bout en bout:

- La messagerie Tutanota qui utilise le chiffrement de bout en bout, Tutanota chiffre localement les messages dans le navigateur avec une méthode hybride standard constituée d'une clé asymétrique RSA 2048 bits et d'une autre symétrique AES 128 bits.
- La messagerie Protonmail qui utilise le chiffrement de bout en bout, ProtonMail utilisé pour le chiffrement des certificats SSL en RSA-4096 (4096 bits) avec hachage grâce à l'algorithme SHA256

5) Récemment, différents projets de loi et règlements (CSAR, EARN IT Act) visent à inciter voire obliger les fournisseurs de services numériques à pouvoir déchiffrer (et donc analyser) les communications de leur.e.s utilisateur.rices. Discutez des arguments en faveur ou contre ces législations, notamment en matière de vie privée.

Les loi et règlements partent d'une bonne intention qui est de détecter des contenus d'abus sexuels de mineurs en analysant les conversations de leurs utilisateurs. Le problème de cette loi est que plus aucune de nos communications en ligne restera privée, tous nos messages seront analysés et lus par des tiers personnes, ce qui va à l'encontre du respect de notre intimité. Cela peut créer un espionnage de masse sur la société, ce qui est néfaste. Le cryptage est une bonne chose cela permet de communiquer avec des personnes choisies sans qu'aucune information fuitent, vouloir garder nos conversations privées n'est pas une mauvaise chose. Le problème du cryptage c'est qu'il y a des abue, certaine personne l'utilise à mauvais escient, c'est pour cela que différents projet de loi apparaissent comme CSAR et

EARN IT pour lutter contre les abus sexuel. Mais selons nous ces lois sont trop abusif et n’ont pas seulement pour but de lutter contre les abus sexuel.

Répartition des tâches:

- Pour l'échauffement nous avons tout fait à deux.
- Pour la partie 1 Arthur a répondu au question et a fait les graphes, Romain a, quant à lui, fait le double SDES ainsi que le cassage brutal. Le cassage astucieux a lui été fait à deux.
- Pour la partie 2 Arthur a fait le cryptage AES et les graphes Romain à répondu aux questions et a fait le décryptage du message caché dans l’image.
- Enfin les parties 3 et 4 ont entièrement été faites à deux.